IMAGE RECOGNITION WITH IBM CLOUD VISUAL RECOGNITION

PROBLEM DEFINITION

Then goal is to develop a platform where users can upload images, and the system accurately classifies and describes the image contents. This will enable users to craft engaging visual storing with the help of AI – Generated captions, The project involves creating an image recognition system using IBM cloud visual recognition. enhancing their connection with the audience through captivating visuals and compelling narratives.

DESIGN THINKING:

Image recognition setup:

To set up the IBM Cloud Visual Recognition service and obtain the API key, you can follow these steps:

1. Go to the IBM Cloud website and sign in to your account.
2. Navigate to the Catalog and search for "Visual Recognition" in the search bar.
3. Select the Visual Recognition service from the results.
4. Follow the prompts to create an instance of the service.
5. Once the instance is created, you'll be able to access the API key in the service credentials section.

User interface:

For a user-friendly interface to upload images and view AI-generated captions:

1. Create a simple and intuitive interface with a "Upload Image" button prominently displayed.

2. Once the user clicks on the button, they can choose an image from their device or take a photo using their camera.

3. After the image is uploaded, display a loading spinner or progress bar to indicate that the AI is generating the caption.

4. Once the caption is ready, show it on the screen along with the uploaded image.

5. Include a "Refresh" or "Upload another Image" button to allow users to easily upload and generate captions for more images.

6. Consider adding a share button to allow users to share the image and caption on social media platforms.

Image classification:

I have written below code in Node.js to create a classifier and then classify an image using **Visual Recognition API ,** but the image isn't classified with the created classifier.
Same code has worked with earlier version.

```
var visual_recognition, params;
var ONE_HOUR = 3600000;
var CLASSIFIERID = [];


   // Create the service wrapper
  visual_recognition = watson.visual_recognition({
    version: 'v3',
    api_key: process.env.API_KEY || '<api-key>',
    version_date: '2015-05-19'
   })
```

Creating classfier:

```
params = {
   name: constants.DRIVERNAME,
   driverOne_positive_examples:
fs.createReadStream('./public/positive.zip'),
```

```
  negative_examples: fs.createReadStream('./public/negative.zip')
  };

  visual_recognition.createClassifier(params, function(err,
classifier) {
   if (err){

     res.render('showError',{title:constants.TITLE1,
                        err:'Something went wrong!'
                        });
   }

   else{

     CLASSIFIERID.push(classifier.classifier_id);
      }

  });
```

Classify image:

```
var parm = {
    images_file: img_classify,
    classifier_ids: CLASSIFIERID,
    threshold: 0.0
  };

visual_recognition.classify(parm, function(err, results) {
  var driverName,driverScore,driverId,driver;

  if (err){
    console.log('Error at classification!!!');

  }

  else{
```

```
      console.log('Image has been classified!!!');
   res.json(results);
}
```

AI generated captions:

The IBM Cloud Visual Recognition API uses advanced natural language processing to generate great captions for recognized images.

It analyses the content of the image and provides descriptive and contextually relevant captions. It's a powerful tool for adding meaningful captions to images.

User Engagement :

To engage users and allow them to share their AI-enhanced images, you can incorporate the following features:

1. AI-Enhanced Filters: Develop a variety of filters that utilize AI to enhance images with unique effects, styles, or themes. Users can apply these filters to their photos and see their images transformed.

2. Caption Generator: Integrate an AI-powered caption generator that automatically generates creative and contextual captions for images. Users can choose to use these captions or customize them to their liking.

3. Social Sharing: Implement social sharing functionality that allows users to easily share their AI-enhanced images on popular social media platforms like Instagram, Facebook, or Twitter. This way, users can showcase their creations with friends and followers.

4. In-App Community: Create an in-app community where users can discover and interact with each other's AI-enhanced images. Users can like, comment, and follow other users, fostering a sense of community and encouraging engagement.

5. Image Challenges: Organize image challenges or contests where users can submit their AI-enhanced images based on specific themes or prompts. This adds a fun and competitive element to the experience and encourages users to showcase their creativity.

Remember to prioritize user privacy and provide clear options for users to control the visibility and sharing settings of their AI-enhanced images.

***